CREDIT CARDS CUSTUMERS

FELIPE

13/12/2020

APLICAÇÃO DE MACHINE LEARNING EM BASE DE DADOS *CREDIT CARDS CUSTUMERS*

A baseado nas variaveis da base de dados BankChurners, criaremos um modelo

de machine learning para prever se cancelaram o cartão ou continurão a ser clientes.

Nosso atributo previsor será Attrition_Flag

Bibliotecas utilizadas

```
library(tidyverse)
library(dplyr)
library(tidyr)
library(readxl)
library(stringr)
library(lubridate)
library(na.tools)
library(data.table)
library(caTools)
library(caret)
library(randomForest)
```

Importando a base de dados BankChurners

```
base_bankChuners<-read_csv("BankChurners.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Attrition_Flag = col_character(),
##   Gender = col_character(),
##   Education_Level = col_character(),
##   Marital_Status = col_character(),
##   Income_Category = col_character(),
##   Card_Category = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```r
base_bankChuners
```

```
## # A tibble: 10,127 x 23
##    CLIENTNUM Attrition_Flag Customer_Age Gender Dependent_count Education_Level
##        <dbl> <chr>                 <dbl> <chr>            <dbl> <chr>
##  1 768805383 Existing Cust~           45 M                    3 High School
##  2 818770008 Existing Cust~           49 F                    5 Graduate
##  3 713982108 Existing Cust~           51 M                    3 Graduate
##  4 769911858 Existing Cust~           40 F                    4 High School
##  5 709106358 Existing Cust~           40 M                    3 Uneducated
##  6 713061558 Existing Cust~           44 M                    2 Graduate
##  7 810347208 Existing Cust~           51 M                    4 Unknown
##  8 818906208 Existing Cust~           32 M                    0 High School
##  9 710930508 Existing Cust~           37 M                    3 Uneducated
## 10 719661558 Existing Cust~           48 M                    2 Graduate
## # ... with 10,117 more rows, and 17 more variables: Marital_Status <chr>,
## #   Income_Category <chr>, Card_Category <chr>, Months_on_book <dbl>,
## #   Total_Relationship_Count <dbl>, Months_Inactive_12_mon <dbl>,
## #   Contacts_Count_12_mon <dbl>, Credit_Limit <dbl>, Total_Revolving_Bal <dbl>,
## #   Avg_Open_To_Buy <dbl>, Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>,
## #   Total_Trans_Ct <dbl>, Total_Ct_Chng_Q4_Q1 <dbl>,
## #   Avg_Utilization_Ratio <dbl>,
## #   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
## #   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
```

PRÉ - PROCESSAMENTO

Excluindo variaveis que não serão utulizadas

```r
base_bankChuners[,c(1,22,23)]
```

```
## # A tibble: 10,127 x 3
##    CLIENTNUM Naive_Bayes_Classifier_Attrition~ Naive_Bayes_Classifier_Attrition~
##        <dbl>                             <dbl>                             <dbl>
##  1 768805383                         0.0000934                              1.00
##  2 818770008                         0.0000569                              1.00
##  3 713982108                         0.0000211                              1.00
##  4 769911858                         0.000134                               1.00
##  5 709106358                         0.0000217                              1.00
##  6 713061558                         0.0000551                              1.00
##  7 810347208                         0.000123                               1.00
##  8 818906208                         0.0000858                              1.00
##  9 710930508                         0.0000448                              1.00
## 10 719661558                         0.000303                               1.00
## # ... with 10,117 more rows
```

```r
base_bankChuners<-base_bankChuners[,-c(1,22,23)]
```

Como ficou.

```r
base_bankChuners
```

```
## # A tibble: 10,127 x 20
##    Attrition_Flag Customer_Age Gender Dependent_count Education_Level
##    <chr>                 <dbl> <chr>            <dbl> <chr>
##  1 Existing Cust~           45 M                    3 High School
##  2 Existing Cust~           49 F                    5 Graduate
##  3 Existing Cust~           51 M                    3 Graduate
##  4 Existing Cust~           40 F                    4 High School
##  5 Existing Cust~           40 M                    3 Uneducated
##  6 Existing Cust~           44 M                    2 Graduate
##  7 Existing Cust~           51 M                    4 Unknown
##  8 Existing Cust~           32 M                    0 High School
##  9 Existing Cust~           37 M                    3 Uneducated
## 10 Existing Cust~           48 M                    2 Graduate
## # ... with 10,117 more rows, and 15 more variables: Marital_Status <chr>,
## #   Income_Category <chr>, Card_Category <chr>, Months_on_book <dbl>,
## #   Total_Relationship_Count <dbl>, Months_Inactive_12_mon <dbl>,
## #   Contacts_Count_12_mon <dbl>, Credit_Limit <dbl>, Total_Revolving_Bal <dbl>,
## #   Avg_Open_To_Buy <dbl>, Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>,
## #   Total_Trans_Ct <dbl>, Total_Ct_Chng_Q4_Q1 <dbl>,
## #   Avg_Utilization_Ratio <dbl>
```

Dimensão de nosso banco de dados.

10127 linhas e 20 colunas

```r
dim(base_bankChuners)
```

```
## [1] 10127    20
```

MOVENDO ATRIBUTO PREVISOR PARA ULTIMA COLUNA.

```r
base_bankChuners<-base_bankChuners[,c(2:20,1)]

base_bankChuners
```

```
## # A tibble: 10,127 x 20
##    Customer_Age Gender Dependent_count Education_Level Marital_Status
##           <dbl> <chr>            <dbl> <chr>           <chr>
##  1           45 M                    3 High School     Married
##  2           49 F                    5 Graduate        Single
##  3           51 M                    3 Graduate        Married
##  4           40 F                    4 High School     Unknown
##  5           40 M                    3 Uneducated      Married
##  6           44 M                    2 Graduate        Married
##  7           51 M                    4 Unknown         Married
```

```
## 8            32 M                    0 High School     Unknown
## 9            37 M                    3 Uneducated      Single
## 10           48 M                    2 Graduate        Single
## # ... with 10,117 more rows, and 15 more variables: Income_Category <chr>,
## #   Card_Category <chr>, Months_on_book <dbl>, Total_Relationship_Count <dbl>,
## #   Months_Inactive_12_mon <dbl>, Contacts_Count_12_mon <dbl>,
## #   Credit_Limit <dbl>, Total_Revolving_Bal <dbl>, Avg_Open_To_Buy <dbl>,
## #   Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>, Total_Trans_Ct <dbl>,
## #   Total_Ct_Chng_Q4_Q1 <dbl>, Avg_Utilization_Ratio <dbl>,
## #   Attrition_Flag <chr>
```

Nomes das variaveis com letras maiúsculas.

```r
names(base_bankChuners)<-str_to_upper(names(base_bankChuners))
names(base_bankChuners)
```

```
##  [1] "CUSTOMER_AGE"             "GENDER"
##  [3] "DEPENDENT_COUNT"          "EDUCATION_LEVEL"
##  [5] "MARITAL_STATUS"           "INCOME_CATEGORY"
##  [7] "CARD_CATEGORY"            "MONTHS_ON_BOOK"
##  [9] "TOTAL_RELATIONSHIP_COUNT" "MONTHS_INACTIVE_12_MON"
## [11] "CONTACTS_COUNT_12_MON"    "CREDIT_LIMIT"
## [13] "TOTAL_REVOLVING_BAL"      "AVG_OPEN_TO_BUY"
## [15] "TOTAL_AMT_CHNG_Q4_Q1"     "TOTAL_TRANS_AMT"
## [17] "TOTAL_TRANS_CT"           "TOTAL_CT_CHNG_Q4_Q1"
## [19] "AVG_UTILIZATION_RATIO"    "ATTRITION_FLAG"
```

Transformando atributos previsores em valores categoricos.

```r
base_bankChuners$GENDER<-as_factor(base_bankChuners$GENDER)
base_bankChuners$EDUCATION_LEVEL<-as_factor(base_bankChuners$EDUCATION_LEVEL)
base_bankChuners$MARITAL_STATUS<-as_factor(base_bankChuners$MARITAL_STATUS)
base_bankChuners$INCOME_CATEGORY<-as_factor(base_bankChuners$INCOME_CATEGORY)
base_bankChuners$CARD_CATEGORY<-as_factor(base_bankChuners$CARD_CATEGORY)
base_bankChuners$ATTRITION_FLAG<-as_factor(base_bankChuners$ATTRITION_FLAG) ## atributo previsor
```

Padronizaçao de valores numericos.

A função scale() utiliza a técnica de padronizaçao(Padronization) para os valores numericos, como existem valores muito diferentes... alguns algoritmos podem dar um peso maior para valores numericos de maior valor, como algoritmo KNN que é baseado em distâncias.

sumarização dos atributos numéricos

```r
summary(base_bankChuners[,c(1,3,8:19)])
```

```
##   CUSTOMER_AGE   DEPENDENT_COUNT MONTHS_ON_BOOK  TOTAL_RELATIONSHIP_COUNT
##  Min.   :26.00   Min.   :0.000   Min.   :13.00   Min.   :1.000
##  1st Qu.:41.00   1st Qu.:1.000   1st Qu.:31.00   1st Qu.:3.000
##  Median :46.00   Median :2.000   Median :36.00   Median :4.000
##  Mean   :46.33   Mean   :2.346   Mean   :35.93   Mean   :3.813
##  3rd Qu.:52.00   3rd Qu.:3.000   3rd Qu.:40.00   3rd Qu.:5.000
##  Max.   :73.00   Max.   :5.000   Max.   :56.00   Max.   :6.000
##  MONTHS_INACTIVE_12_MON CONTACTS_COUNT_12_MON  CREDIT_LIMIT
##  Min.   :0.000          Min.   :0.000         Min.   : 1438
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555
##  Median :2.000          Median :2.000         Median : 4549
##  Mean   :2.341          Mean   :2.455         Mean   : 8632
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068
##  Max.   :6.000          Max.   :6.000         Max.   :34516
##  TOTAL_REVOLVING_BAL AVG_OPEN_TO_BUY TOTAL_AMT_CHNG_Q4_Q1 TOTAL_TRANS_AMT
##  Min.   :   0        Min.   :    3   Min.   :0.0000       Min.   :  510
##  1st Qu.: 359        1st Qu.: 1324   1st Qu.:0.6310       1st Qu.: 2156
##  Median :1276        Median : 3474   Median :0.7360       Median : 3899
##  Mean   :1163        Mean   : 7469   Mean   :0.7599       Mean   : 4404
##  3rd Qu.:1784        3rd Qu.: 9859   3rd Qu.:0.8590       3rd Qu.: 4741
##  Max.   :2517        Max.   :34516   Max.   :3.3970       Max.   :18484
##  TOTAL_TRANS_CT   TOTAL_CT_CHNG_Q4_Q1 AVG_UTILIZATION_RATIO
##  Min.   : 10.00   Min.   :0.0000      Min.   :0.0000
##  1st Qu.: 45.00   1st Qu.:0.5820      1st Qu.:0.0230
##  Median : 67.00   Median :0.7020      Median :0.1760
##  Mean   : 64.86   Mean   :0.7122      Mean   :0.2749
##  3rd Qu.: 81.00   3rd Qu.:0.8180      3rd Qu.:0.5030
##  Max.   :139.00   Max.   :3.7140      Max.   :0.9990
```

Padronização (escalonamneto)

```r
base_bankChuners[,c(1,3,8:19)]<-scale(base_bankChuners[,c(1,3,8:19)])
```

sumarização dos atributos numéricos(já padronizados)

```r
summary(base_bankChuners[,c(1,3,8:19)])
```

```
##   CUSTOMER_AGE      DEPENDENT_COUNT   MONTHS_ON_BOOK
##  Min.   :-2.53542   Min.   :-1.8063   Min.   :-2.870926
##  1st Qu.:-0.66435   1st Qu.:-1.0364   1st Qu.:-0.617099
##  Median :-0.04066   Median :-0.2665   Median : 0.008964
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000
##  3rd Qu.: 0.70777   3rd Qu.: 0.5033   3rd Qu.: 0.509814
##  Max.   : 3.32726   Max.   : 2.0431   Max.   : 2.513216
##  TOTAL_RELATIONSHIP_COUNT MONTHS_INACTIVE_12_MON CONTACTS_COUNT_12_MON
##  Min.   :-1.8094          Min.   :-2.3166        Min.   :-2.2195
##  1st Qu.:-0.5228          1st Qu.:-0.3376        1st Qu.:-0.4116
##  Median : 0.1206          Median :-0.3376        Median :-0.4116
##  Mean   : 0.0000          Mean   : 0.0000        Mean   : 0.0000
##  3rd Qu.: 0.7639          3rd Qu.: 0.6519        3rd Qu.: 0.4924
##  Max.   : 1.4072          Max.   : 3.6204        Max.   : 3.2043
##   CREDIT_LIMIT     TOTAL_REVOLVING_BAL AVG_OPEN_TO_BUY   TOTAL_AMT_CHNG_Q4_Q1
##  Min.   :-0.7915   Min.   :-1.4268     Min.   :-0.8213   Min.   :-3.4668
```

```
##  1st Qu.:-0.6686   1st Qu.:-0.9863     1st Qu.:-0.6759   1st Qu.:-0.5882
##  Median :-0.4492   Median : 0.1389     Median :-0.4395   Median :-0.1092
##  Mean   : 0.0000   Mean   : 0.0000     Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2680   3rd Qu.: 0.7622     3rd Qu.: 0.2629   3rd Qu.: 0.4519
##  Max.   : 2.8479   Max.   : 1.6616     Max.   : 2.9752   Max.   :12.0300
##  TOTAL_TRANS_AMT    TOTAL_TRANS_CT     TOTAL_CT_CHNG_Q4_Q1
##  Min.   :-1.14629   Min.   :-2.33714   Min.   :-2.99145
##  1st Qu.:-0.66191   1st Qu.:-0.84604   1st Qu.:-0.54695
##  Median :-0.14868   Median : 0.09123   Median :-0.04294
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.09918   3rd Qu.: 0.68767   3rd Qu.: 0.44428
##  Max.   : 4.14465   Max.   : 3.15864   Max.   :12.60795
##  AVG_UTILIZATION_RATIO
##  Min.   :-0.9971
##  1st Qu.:-0.9137
##  Median :-0.3587
##  Mean   : 0.0000
##  3rd Qu.: 0.8274
##  Max.   : 2.6265
```

DIVIDINDO BASE DE DADOS EM TREINAMNETO E TESTE.

```r
library(caTools)

set.seed(1)
dividir<-sample.split(Y = base_bankChuners$ATTRITION_FLAG,SplitRatio = 0.75)
base_treinamento<-subset(x = base_bankChuners,subset = dividir == TRUE)
base_teste<-subset(x = base_bankChuners,subset = dividir == FALSE)
```

TREINANDO MODELO DE ALGORITMO RANDOM FOREST.

```r
library(randomForest)

set.seed(1)                ## set.seed(1)
mdl_Random_Forest<-randomForest(formula = ATTRITION_FLAG ~.,data = base_treinamento,ntree = 90)
```

APLICANDO O MODELO RANDOM FOREST.

```r
previsao<-predict(mdl_Random_Forest,newdata = base_teste[,-20])
```

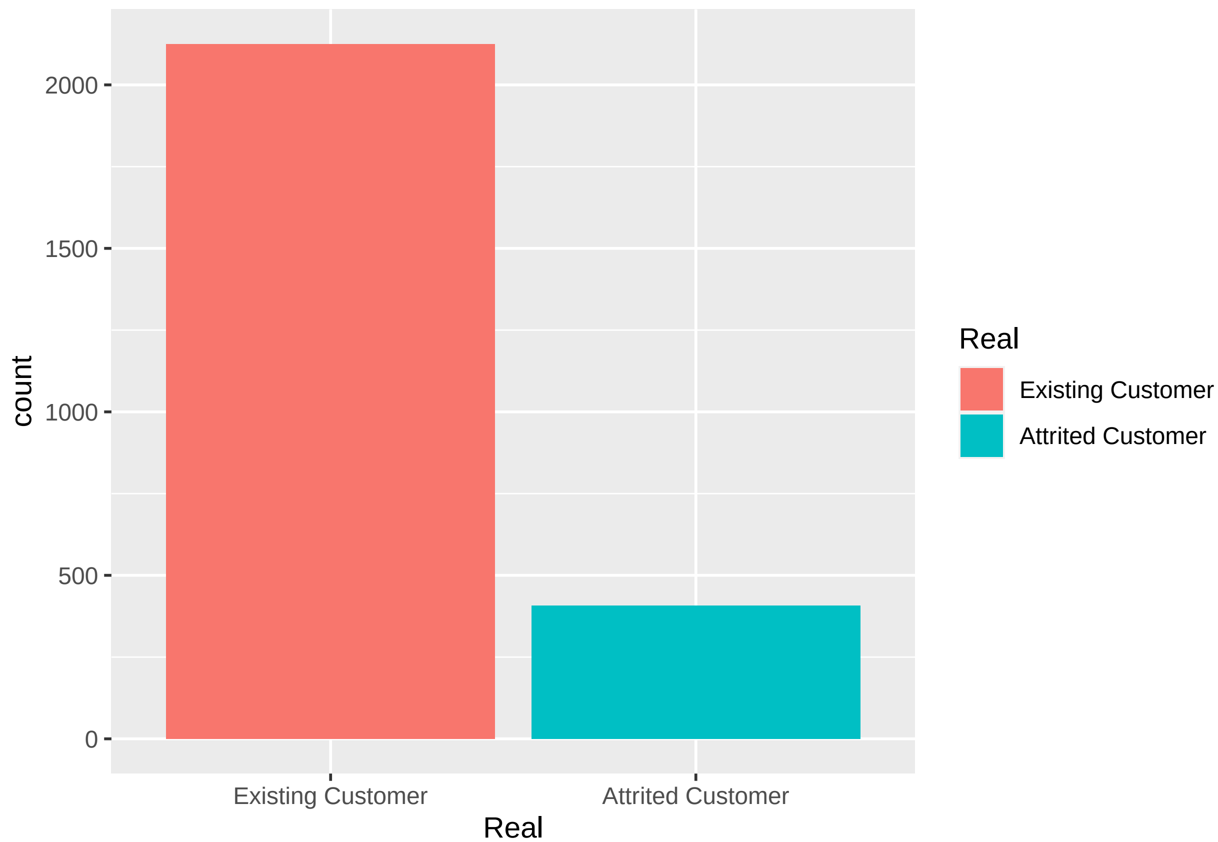
EXIBINDO O RESULTADO DE NOSSA PREVISAO JUNTAMENTE COM OS VALORES DE TESTE.

```r
df<-data.frame(Real=base_teste$ATTRITION_FLAG,Previsao=previsao)
head(df)
```

```
##                   Real          Previsao
## 1 Existing Customer Existing Customer
## 2 Existing Customer Existing Customer
## 3 Existing Customer Existing Customer
## 4 Existing Customer Existing Customer
## 5 Existing Customer Existing Customer
## 6 Existing Customer Existing Customer
```

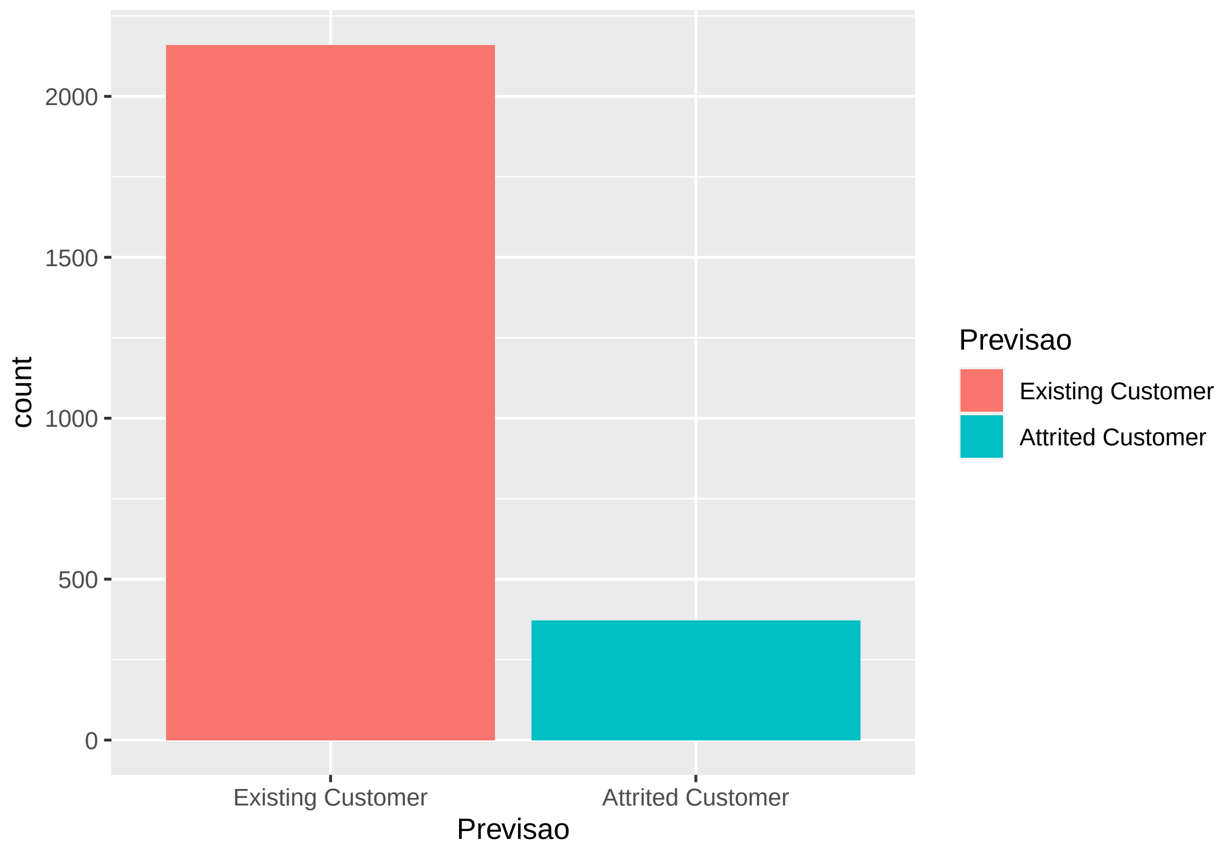
VALORES DO REAIS, BASE TESTE.

```r
ggplot(data = df)+
  geom_bar(mapping = aes(x = Real,fill = Real))
```



VALORES DA PREVISAO

```r
ggplot(data = df)+
  geom_bar(mapping = aes(x = Previsao ,fill = Previsao))
```



MATRIZ DE CONFUSAO PARA VERIFICAR ACERTOS E ERROS DO MODELO.

```
library(caret)
matriz_confusao<-table(base_teste$ATTRITION_FLAG,previsao)

matriz_confusao
```

```
##                     previsao
##                      Existing Customer Attrited Customer
##   Existing Customer               2102                23
##   Attrited Customer                 58               349
```

MATRIZ DE CONFUSAO PARA VERIFICAR A ACURACIDADE DE NOSSO MODELO.

```
confusionMatrix(matriz_confusao)
```

```
## Confusion Matrix and Statistics
##
##                     previsao
##                      Existing Customer Attrited Customer
##   Existing Customer               2102                23
```

```
##   Attrited Customer                58               349
##
##                Accuracy : 0.968
##                  95% CI : (0.9604, 0.9745)
##     No Information Rate : 0.8531
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8772
##
##  Mcnemar's Test P-Value : 0.0001582
##
##             Sensitivity : 0.9731
##             Specificity : 0.9382
##          Pos Pred Value : 0.9892
##          Neg Pred Value : 0.8575
##              Prevalence : 0.8531
##          Detection Rate : 0.8302
##    Detection Prevalence : 0.8393
##       Balanced Accuracy : 0.9557
##
##        'Positive' Class : Existing Customer
##
```

RESULTADO DE NOSSO MODELO DE MACHINE LEARNING RANDOM FOREST.

OBTIVEMOS A ACURRACIDADE DE:

96.8 % - Modelo Random Forest– Nº de arvores do modelo 90.

Utilizamos: valores categoricos + escalonamento de valores numéricos